SOLE INVENTOR

APPLICATION FOR
UNITED STATES LETTERS PATENT

S P E C I F I C A T I O N

TO ALL WHOM IT MAY CONCERN:

Be it known that I, **Robert B. Havekost** a citizen of the United States, residing at **14507 Crystal Court** and State of **Texas** have invented a new and useful **ENHANCED DEVICE ALARMS IN A PROCESS CONTROL SYSTEM**, of which the following is a specification.

ENHANCED DEVICE ALARMS
IN A PROCESS CONTROL SYSTEM

RELATED APPLICATION

This application is a continuation-in-part of U.S. Patent Application Serial No.

5   09/707,580 entitled "Integrated Alarm Display in a Process Control Network," which

was filed on November 7, 2000.

FIELD OF THE INVENTION

The present invention relates generally to process control systems and, more

particularly, to the enhancement and display of device alarms within a process control

10   system.

DESCRIPTION OF THE RELATED ART

Process control systems, like those used in chemical, petroleum or other

processes, typically include one or more centralized process controllers

communicatively coupled to at least one host or operator workstation and to one or

15   more field devices via analog, digital or combined analog/digital buses. The field

devices, which may be, for example valves, valve positioners, switches and

transmitters (e.g., temperature, pressure and flow rate sensors), perform functions

within the process such as opening or closing valves and measuring process

parameters. The process controller receives signals indicative of process

20   measurements made by the field devices and/or other information pertaining to the

field devices, uses this information to implement a control routine and then generates

control signals which are sent over the buses or other communication lines to the field

devices to control the operation of the process. Information from the field devices and

the controllers may be made available to one or more applications executed by the

25   operator workstation to enable an operator to perform desired functions with respect

to the process, such as viewing the current state of the process, modifying the

operation of the process, etc. Some known controller/operator interface software is

designed to generate and display process alarms resulting from process control

operations performed by software in the controllers or other devices.

30   The DeltaV process control system sold by Fisher Rosemount Systems, Inc.

uses function blocks located or installed in controllers or different field devices to

perform control operations. The controllers and, in some cases, the field devices are capable of storing and executing one or more function blocks, each of which receives inputs from and/or provides outputs to other function blocks (either within the same device or within different devices), and performs some process control operation, such as measuring or detecting a process parameter, controlling a device or performing a control operation, such as implementing a proportional-derivative-integral (PID) control routine. The different function blocks within a process control system are configured to communicate with each other (e.g., within a single device or over a bus) to form one or more process control loops, the individual operations of which can thus be distributed throughout the process control system.

Typically, the function blocks or the devices in which these function blocks are implemented are configured to detect errors, faults or problems that occur within the process control loops or the functions being performed therein and to send a signal, such as an alarm message, to notify an operator at an operator workstation or other user interface that an undesirable condition exists within the process control system or a control loop of the process control system. Such alarms may indicate, for example, that a function block is not communicating, that a function block has received or generated an out of range input or output, that a function block is undergoing a fault or other undesirable condition, etc. In current alarm processing and display systems, an application executed at, for example, an operator interface/workstation, is configured to receive messages containing process alarms related to process operation and to display these process alarms in a coherent and manageable manner to thereby enable an operator to manage alarms in some organized or logical way. Such an operator interface system is described in U.S. Patent No. 5,768,119, entitled "Process Control System Including Alarm Priority Adjustment," which is hereby expressly incorporated by reference herein.

Known operator interface applications, such as that described in U.S. Patent No. 5,768,119, are typically configured to enable an operator, i.e., the person overseeing the actual day-to-day operation of a process control system, to view the most critical process alarms, such as the alarms with the highest priority, first.

Because these applications are designed to provide information to a process control operator, these applications only display alarms associated with the functioning of the process itself and are not configured to display other types of errors or alarms, such as alarms associated with malfunctioning field devices or other hardware such as

5   controllers or input/output (I/O) devices. Thus, for example, in the system described in U.S. Patent No. 5,768,119, an operator display application displays a section of a process control system and provides an alarm banner on the bottom of the display indicating the highest priority process alarms. The displayed alarms are process alarms because they are generated by function blocks or other software used to

10   implement a process control scheme or a process control loop and to indicate an error in the functioning of a process control loop. When an operator selects one of the process alarms at the operator workstation, the application provides the operator more information related to the selected alarm, such as the function block or module which generated the alarm, the priority of the alarm, whether the alarm has been

15   acknowledged, etc. and may display information about the process relevant to the alarm, such as a faceplate for the loop in which the alarm occurred, a primary control display related to the portion of the plant in which the alarm occurred, etc.

Using these known displays, an operator can recognize the existence of an alarm and may try to correct the problem using other available software applications.

20   In some cases, the operator may determine, based on the process alarms present, that one or more field devices or other hardware components may need repair or replacement. If this is the case, the operator may contact a maintenance person to schedule the device for testing, repair or replacement as needed. Similarly, the operator may contact an engineer to respond to the alarm. In this manner, the operator

25   detects a device or hardware problem based on received process alarms and manually passes the problem off to maintenance or engineer personnel to correct the problem. However, it requires an experienced or knowledgeable operator to detect or deduce device or hardware problems from process alarms.

In the past, conventional field devices were used in process control systems to

30   send and receive analog signals, such as, for example, 4-20 milliamp (mA) signals to

and from the process controller via an analog bus or analog lines. However, these 4-20 mA signals are limited in nature because they are only indicative of process measurements made by the device or of process control signals generated by the controller required to control the operation of the device during runtime. As a result,

5   conventional 4-20 mA devices are incapable of generating alarms pertaining to the operational capability or status of the device itself. As a result, alarms associated with these devices have generally not been available within process control systems. However, more recently, smart field devices including a microprocessor and a memory have become prevalent in the process control industry. A number of standard

10   and open smart device communication protocols such as the FOUNDATION™ Fieldbus (hereinafter "Fieldbus"), HART®, PROFIBUS®, WORLDFIP®, Device-Net®, and CAN protocols, have been developed to enable smart field devices made by different manufacturers to be used together within the same process control network. In addition to performing a primary function within the process, a smart field device may

15   store data pertaining to the device, communicate with the controller and/or other devices in a digital or combined digital and analog format and may perform secondary tasks such as self-calibration, identification, diagnostics, etc. Importantly, the devices conforming to at least some of these protocols are capable of detecting problems within the device itself and of generating and sending alarm messages or alert

20   messages to indicate the detected problems to the appropriate operator, maintenance or engineer personnel responsible for the proper operation of the process control system.

However, there have been few if any display applications for displaying non-process alarms, such as alarms generated by the field devices or controllers indicating

25   that some problem with the hardware associated with those devices has occurred. It is believed that there are no known displays which enable maintenance or engineer personnel to detect and attend to faulty field devices, controllers, I/O devices, etc. in a coherent or consistent manner, as there are with operator displays that display process alarms. In fact, engineers and technical personnel generally have to review numerous

30   log printouts or stored files for alarms or other events detected about or by field

devices or other hardware devices. This process is time consuming and difficult. The capability of enabling the engineer or maintenance person to view device and hardware alarms in a coherent manner is even more desirable with the advent of newer smart field devices, controllers and I/O devices which, as indicated above,

5   provide the capability to detect numerous operating problems about the devices themselves, such as when a device stops communicating or some other problem that prevents a device from operating correctly. For example, a valve may include sensors that detect an overpressure or an underpressure condition, a stuck valve plug, etc. and software within the device may result in the generation of an alarm message indicating

10   the type of detected problem as well as the severity of the problem. In some cases, the problem may be critical and may need immediate attention, such as with a stuck valve plug, whereas other conditions may require scheduling maintenance at some time in the future. For example, a valve may measure aggregate valve travel since the last maintenance and, when the aggregate valve travel reaches a predetermined threshold,

15   the smart valve device may generate an alarm calling for maintenance.

Still further, in some smaller process control applications, one person may want to perform the functions of an operator, a maintenance person and an engineer. Alternatively, the operator may wish to know about problems occurring in field devices, controllers and I/O devices because it may help the operator make sense of

20   the process alarms that appear within the traditional operator interface. In the past, there was no easy way of enabling a single person to view and interpret all of the alarm or event data in an organized and integrated manner. Still further, an operator may wish to view all categories of alarms, unless too much information is being displayed on the interface, at which time the operator may wish to turn certain

25   categories of alarms off or hand the responsibility for certain alarms to someone else. Currently, there is no system which enables a single person to perform all of these functions.

SUMMARY OF THE INVENTION

An alarm display and interface system for use in a process control system

30   enhances device alarm messages to enable the integrated display of device alarms with

process alarms, thereby facilitating an operator, user, etc. in managing and responding to multiple categories of alarms within a single interface.

In accordance with one aspect of the invention, a system and method of processing an alarm within a process control system may receive a device alarm

5   message from a field device and the device alarm message may have a first set of message fields including a first set of device information. The system and method may append a second set of message fields containing a second set of device information associated with the field device to the device alarm message to form an enhanced device alarm message and may select a device alarm table based on the

10   second set of device information within the enhanced device alarm message.

Additionally, the system and method may map the first set of device information within the enhanced device alarm message to the selected device alarm table to identify a set of information fields containing displayable alarm information and may display a portion of the displayable alarm information including one of an

15   alarm parameter, an alarm description, an alarm type, an alarm message and corrective action information. Still further, the system and method may display a floating point numeric value from one of the first set of message fields in response to a predetermined character string within one of the set of information fields containing displayable alarm information.

20   In accordance with another aspect of the invention, a system for processing a device alarm within a process control network having a field device includes a processor system communicatively coupled to a memory. The processor system may be programmed to receive a device alarm message from the field device and the device alarm message may have a first set of device information stored within the

25   field device. The processor system may be further programmed to append a second set of device information associated with the field device to the device alarm message to form an extended device alarm message and may use the first set of device information within the extended device alarm message to identify a set displayable alarm information.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a process control system in which an integrated alarm display and interface system may be used;

Fig. 2 is a block diagram of a workstation having an integrated alarm display and interface system executed therein;

Fig. 3 is a first exemplary user interface screen generated by the integrated alarm display and interface system used in the process control system of Fig. 1;

Fig. 4 is a second exemplary user interface screen generated by the integrated alarm display and interface system used in the process control system of Fig. 1;

Fig. 5 is an exemplary alarm banner display for a process alarm;

Fig. 6 is an exemplary alarm banner display for a device alarm;

Fig. 7 is an exemplary alarm banner display for a hardware alarm;

Fig. 8 is an exemplary faceplate display for a device alarm associated with a Fieldbus field device;

Fig. 9 is an exemplary faceplate display for a device alarm associated with a generic field device;

Fig. 10 is an exemplary faceplate display for a hardware alarm associated with a node or a hardware element;

Fig. 11 is an exemplary interface display for use in setting the categories and priorities of alarms to be displayed on a user interface screen;

Fig. 12 is a first exemplary alarm summary display created by an alarm summary control module;

Fig. 13 is a second exemplary alarm summary display created by an alarm summary control module;

Fig. 14 is a third exemplary alarm summary display created by an alarm summary control module; and

Fig. 15 is an exemplary event chronicle for process and device alarms.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring now to Fig. 1, a process control network or system 10 includes one or more process controllers 12 connected to one or more host workstations or

computers 14 (which may be any type of personal computer or workstation) and banks

of input/output (I/O) devices 20, 22, each of which is connected to one or more field

devices 25-39. The controllers 12 may be, for example, DeltaV™ controllers sold by

Fisher-Rosemount Systems, Inc., and are communicatively connected to the host

5   computers 14 via, for example, an Ethernet connection 40 or any other suitable

communication link. Likewise, the controllers 12 are communicatively connected to

the field devices 25-39 using any desired hardware and software associated with, for

example, standard 4-20 mA devices and/or any smart communication protocol such as

the Fieldbus or HART protocols. As is generally known, the controllers 12

10   implement or supervise process control routines stored therein or otherwise associated

therewith and communicate with the field devices 25-39 to control a process in any

desired manner.

        The field devices 25-39 may be any types of devices, such as sensors, valves,

transmitters, positioners, etc., while the I/O cards within the banks 20 and 22 may be

15   any types of I/O devices conforming to any desired communication or controller

protocol such as HART, Fieldbus, Profibus, etc. In the embodiment illustrated in Fig.

1, the field devices 25-27 are standard 4-20 mA devices that communicate over analog

lines to the I/O card 22A, the field devices 28-31 are illustrated as HART devices

connected to a HART compatible I/O device 20A, and the field devices 32-39 are

20   smart devices, such as Fieldbus field devices, that communicate over digital bus 42 or

44 to the I/O cards 20B or 22B using, for example, Fieldbus protocol

communications. Of course, the field devices 25-39 and the banks of I/O cards 20 and

22 may conform to any other desired standards or protocols instead of the 4-20 mA,

HART or Fieldbus protocols, including any standards or protocols developed in the

25   future.

        Each of the controllers 12 is configured to implement a control strategy using

what are commonly referred to as function blocks. As is well known, each function

block is a part (e.g., a subroutine) of an overall control routine and operates in

conjunction with other function blocks (via communications called links) to

30   implement process control loops within the process control system 10. Function

blocks typically perform an input function, such as that associated with a sensor or other process parameter measurement device, a control function, such as that associated with a control routine that performs PID control, fuzzy logic control, etc., or an output function that controls the operation of some device, such as a valve, to

5   perform some physical function within the process control system 10. Of course, hybrid and other types of function blocks exist. Groups of function blocks are called modules. Function blocks and modules may be stored in and executed by the controller 12, which is typically the case when function blocks are used for, or are associated with, standard 4-20 mA devices and some types of smart field devices, or

10   may be stored in and implemented by the field devices themselves, which may be the case with Fieldbus devices. While the description of the control system 10 is provided herein using a function block control strategy, the control strategy could also be implemented using other techniques, such as ladder logic, sequential flow charts, etc. and using any desired proprietary or non-proprietary programming language.

15   In the system of Fig. 1, one or more of the host devices 14 functions as an operator workstation and has alarm processing software 50 stored therein. Generally speaking, the alarm processing software 50 displays information about the process control system 10 pertinent to the operator's understanding or ability to view the current operational status of the process with respect to the alarms present in the

20   system. For example, the alarm processing software 50 may display an alarm banner having alarm indications therein and a primary control display illustrating a section of the process control system 10, including the devices and other equipment associated with that section of the process control system 10 relevant to one of the alarms. The primary control display may provide information about the current state of the process

25   control system 10, such as the level of a fluid in a tank, the flow characteristic of a valve and other fluid lines, the settings of equipment, the readings of sensors, etc. An example of such a display is illustrated in Fig. 3. An operator may use the alarm processing software 50 to view different parts of the process control system 10 or equipment within the process control system 10. Of course, the alarm processing

30   software 50 communicates with the controllers 12 and, if necessary, the field devices

25-39, any of the banks of I/O devices 20, 22 or any other devices to obtain the relevant values, settings and measurements associated with or being made in the process control system 10 to create the interface screen on the operator display of the workstation 14.

5   The alarm processing software 50 is configured to receive alarm messages created by alarm generating software within some or all of the controllers 12, the I/O devices 20 and 22 and/or the field devices 25-39. This software is illustrated as software elements 51, 52 and 53 in Fig. 1. Generally speaking, the alarm processing software 50 receives different categories of alarm messages including, for example,

10   process alarms (which are typically generated by a process control software modules, such as those made up of communicatively interconnected function blocks, forming process control routines used during runtime of the process), hardware alarms, such as alarms generated by the controllers 12, I/O devices 20 and 22 or other workstations 14, pertaining to the state or functioning condition of these devices, and device

15   alarms, which are generated by some or all of the field devices 25-39 to indicate problems associated with those devices. These or other categories of alarms may be generated in any desired manner. For example, it is well known to have the function blocks or software modules that are used to implement process control functions generate process alarms, and these process alarms are typically sent in the form of

20   alarm messages to operator interfaces for display. Also, newer smart devices, controllers, I/O devices, databases, servers, workstations, etc. may use any desired proprietary or non-proprietary software to detect problems, errors, maintenance alerts, etc. and may send alarms indicating these conditions to the operator interface 14. In particular, many of the devices now available, such as controllers, I/O devices and

25   smart field devices are provided with software and/or sensors that detect hardware problems, such as a stuck valve plug, broken parts, maintenance concerns, etc. and may generate signals or messages indicting these conditions.

If desired, the alarm processing software 50 may receive and filter alarms based on a number of factors. In particular, the alarm processing software 50 may

30   filter alarms based on the workstation in which the software 50 is executed, the

identity the person logged into the workstation, and operator configurable settings, such as category, type, priority, status, time of generation, etc. of the alarm. For example, the alarm processing software 50 may filter alarms to selectively display alarms from the areas or sections of the plants that the workstation executing the

5   alarm processing software 50 is configured to receive. In other words, alarms for certain areas or sections of the plant may not be displayed at particular workstations but, instead, each workstation may be limited to displaying alarms for one or more specific areas of the plant. Likewise, alarms may be filtered by operator identification. In particular, individual operators may be limited to viewing certain

10   categories, types, priority levels, etc. of alarms or may be limited to viewing alarms from a section or subsection (e.g., an area) of the plant. The alarm processing software 50 may also filter alarms for display based on the operator's security clearance. These workstation and operator filtering settings are referred to herein as workstation and operator scope controls.

15   The alarm processing software 50 may also filter the viewable alarms (i.e., those within the workstation and operator scope controls) based on operator configurable settings including, for example, the alarm category (e.g., process, device or hardware alarm), alarm type (communication, failure, advisory, maintenance, etc.), the alarm priority, the module, device, hardware, node or area to which the alarm

20   pertains, whether the alarm has been acknowledged or suppressed, whether the alarm is active, etc.

Referring now to Fig. 2, the configuration of one of the workstations 14 that implements the alarm display and interface system is illustrated in more detail. As illustrated in Fig. 2, the workstation 14 stores and executes communication software,

25   such as a communication layer or stack 62, that communicates with the controllers 12 via the Ethernet connection 40 to receive signals sent by the controllers 12, I/O devices within the banks 20 and 22, field devices 25-39 and/or other workstations 14. The communication layer 62 also properly formats messages to be sent to the controllers, I/O devices, field devices 25-39 and other workstations 14 such as alarm

30   acknowledgment messages or signals, etc. The communication software used to

- 11 -

implement the communication layer can be any known or desired communication software that is currently used with, for example, Ethernet communications. Of course, the communication stack 62 is coupled to other software that performs other functions, such as configuration applications, diagnostic or other process applications,

5   database management applications, etc. executed within the workstation 14.

The alarm display and interface system includes an alarm processing unit 64 that receives alarms and other event information from the communication layer 62 in the form of messages, decodes those messages containing alarm or other event information and may store the alarm and other event information in a database 66.

10   The front end of the alarm processing unit 64, which interfaces with the communication layer 62 and the database 66, may be an alarm receiver. The alarm processing software 50 also includes an alarm filter 68 that the alarm processing unit 64 uses to determine which alarms are to be displayed on a user interface 69 (such as a CRT, LCD, LED, plasma display, printer, etc.) associated with the workstation 14.

15   The filter 68 may have its settings stored in the database 66 and these filter settings may be preconfigured and/or may be changed by a user based on the user's preferences.

Generally, the filter settings may control the category and priority of alarms and, if desired, may establish the order of the alarms to be displayed using a number

20   of different criteria. The workstation and operator scope controls affect what a particular operator can see (e.g., which alarms can be displayed at a particular workstation) based on the operator identification and workstation to which the operator is logged on. In this case, an operations license may be assigned to each workstation and, without an operations license, the alarm information and all alarm

25   list/summary displays may be empty. In other words, no active or suppressed alarms of any category (i.e., process, hardware or device) will be shown by the alarm processing unit 64. Still further, only alarms from a plant area in the current operator's scope (the operator is usually given at least one security key in the plant area) are eligible to appear in the alarm displays on that workstation. Also, only

30   alarms from a plant area and unit which has not been turned off using the plant area or

unit filtering display(s) (to be discussed below) are eligible to appear in the alarm display. In this manner, the filter 68 prevents the display of alarms outside of the workstation and operator scope and alarms from plant areas or units that have been turned off by the operator.

5   After testing alarms for conformance to the workstation and operator scope controls, the filter 68 filters out and determines the display order of alarms based on operator settings, which may include, for example, the category of alarm, the priority of the alarm, the type of alarm, the acknowledged status of the alarm, the suppressed status of the alarm, the time of the alarm, the active status of the alarm, etc. The

10   received alarms, which are sent to the alarm processing software 50 using alarm messages may include a parameter for each of these values and the filter 68 may filter alarms for display by comparing the appropriate parameters of the alarms to the filter settings. For example, the operator can indicate which categories of alarms and priority levels of alarm should be displayed on the screen. If desired, the operator can

15   adjust a predetermined priority level for an alarm by offsetting the priority level from the preconfigured priority level for the alarm set by the manufacturer. In the DeltaV system, a priority level between about three and fifteen is selected for each alarm and the operator can offset this priority level by any number of levels to make a higher priority a lower priority or a lower priority a higher priority when viewed by the filter

20   68. While the operator may set the order of display of the alarms that are passed by the filter 68, the order may also be determined by preconfigured settings to provide a consistent display of different types of alarms.

In any event, the operator can customize the manner in which alarms are displayed based on the categories or types of alarms that the user is most interested in,

25   which may all be one category or type of alarm such as process alarms, device alarms, or hardware alarms or any combination of two or more categories of alarms. The user may also have control over how the alarms are presented and the information provided with the alarms. In this manner, the alarm processing software 50 enables a single person to perform the operations of an operator, a technician or maintenance person

30   and an engineer by viewing and addressing on the same screen the alarms that would

normally be addressed by different personnel at different locations in a plant. Alternatively, at different times in the same system a maintenance person can use the same system to view only maintenance alarms while an engineer can view other types of alarms that are affecting the devices. In this manner, the alarm processing software

5   50 can be used by different types of people at the same time in different workstations to view different aspects of the alarms associated with the process control system 10. Furthermore, when using the alarm processing software 50, it is relatively easy for an individual to turn over alarm functions that they are viewing and acknowledging to another individual who may have the same software. Alternatively or additionally, an

10   individual may set their filter to accept alarms that are normally viewed by another person. In this manner, one person can go to lunch and turn the alarm viewing function over to other persons at different workstations by resetting a few filter settings. When returning from lunch, that person can regain control of those functions. Also, when the amount of alarm information becomes too large for one

15   person to handle, that person can hand off or shed the load for certain categories of alarms such as process alarms, device alarms or hardware alarms so that these alarms can be handled by other people at other terminals.

After the alarm processing unit 64 uses the filter 68 to decide which alarms should be displayed to the user via the display 69 and the order in which the alarms

20   should be displayed, the alarm processing unit 64 provides this information to a user display interface 70, which uses any standard or desired operating system to display alarm information on the alarm display 69 in any desired manner. Of course, the user display interface 70 obtains other information it needs, such as information about the layout of or the configuration of the process control system 10, the values of

25   parameters or signals within that system, etc. from the database 66 or from other communication signals received from the process control system 10 via the communication layer 62. Also, the user display interface 70 receives commands from the user requesting, for example, more information related to particular alarms, changes to alarm or filter settings, new alarm displays, etc. and provides this

30   information to the processing unit 64, which then takes the requested action, searches

- 14 -

the database 66 for the alarm information, etc. to provide a new alarm view to the user via the display 69.

Generally speaking, there are different categories of alarms that can be generated and displayed on the display 69 including, for example, process alarms, device alarms and hardware alarms. Process alarms, which are known and which are typically generated by function blocks or modules within a process control routine running on a controller or a field device, have, in the past, been sent to and displayed on an operator interface. Process alarms generally indicate a problem with the functional operation of the process control software, i.e., a problem with the process control routine itself such as out of bound measurement, abnormal variances between process parameters and set points, etc. Process alarms are typically configured by the user as components of process control modules and may appear in the configuration information provided on the operator interface as being associated with a module name. Some types of process alarms include bad input/output, out of bound measurements, exceeded thresholds, etc. Because process alarms are well known in the art, they will not be described in more detail herein.

Field devices that communicate using the Fieldbus protocol communicate alarm information using a well known message format. Fieldbus device alarm messages include a block identification field, a relative identification field, a subcode field and a floating point number field. Generally speaking, the fields provided within a Fieldbus device alarm message specify in increasing levels of particularity the source of an alarm message and the nature of the alarm conveyed thereby.

In particular, the block identification field within a device alarm message identifies the block within a Fieldbus device from which the alarm message originated. As is well known, Fieldbus devices may each have one or more associated function blocks, resource blocks and/or transducer blocks that represent various capabilities of that Fieldbus device. For example, a Fieldbus temperature transmitter having two temperature sensing elements may include two transducer blocks (i.e., one for each sensing element) and a function block that reads the outputs of the two sensing elements (via the transducer blocks) to produce an average temperature value.

In any event, a controller, workstation, etc. may use the block identification field within a Fieldbus device alarm message to determine which block generated the alarm message and whether the alarm message was generated by a function block, resource block or a transducer block.

5   The relative identification field of a Fieldbus device alarm message identifies what parameter within a particular block (e.g., a function block, resource block, or transducer block) caused the generation of the alarm message. A given block may have two or more parameters associated with it that can be distinguished from each other by using different values within the relative identification field. For example, a

10   function block may have several inputs and outputs, each of which may be uniquely associated with a different relative identification field value.

The subcode field generally provides a numeric value that is indicative of the nature of the alarm message being transmitted and which is predetermined by the device manufacturer. For example, the subcode field may be used to indicate that a

15   sensor reading is outside of a normal operating range, that a sensor has failed completely, or any other failure which can occur within a Fieldbus device.

In Fieldbus devices the subcode field is device and manufacturer specific so that different types of failures within a particular block of a given Fieldbus device may result in different subcode field values and so that identical types of failures within

20   different devices and/or within similar devices made by different manufacturers may also result in different subcode field values being sent within an alarm message. Because the subcode field is not user configurable and because the subcode field values for particular types of failures are device and/or manufacturer specific, manufacturers typically provide a list of subcodes and corresponding failure types so

25   that the subcode values may be translated by a user, operator, designer, etc. into failure types.

The floating point field typically contains a floating point number that is associated with the subcode being reported within the alarm message. Thus, in the case where a subcode field indicates that a sensor reading within a particular

transducer block is outside of a normal operating range, the floating point field may contain a floating point value representing the actual out of range sensor reading.

Device alarms are alarms associated with the operation of the field devices within the process and may be detected by software (e.g., the software 53 in Fig. 1)

5   within the field devices or other devices connected within the process control system 10 to indicate a problem or error with the operation of a field device. Device alarms may appear in the operator interface of the system described herein as being associated with a particular device. Device alarms may, for example, indicate that the pressure in a valve is to great or to small for proper operation of the valve, that the

10   motor current in the valve is to high or to low, that the voltage levels of a device are not synchronized, that a valve plug within a valve is stuck, that the device is not communicating properly, that the device needs scheduled maintenance because, for example, a certain amount of time has passed or because a valve member of the device has undergone a certain amount of travel since the last maintenance, etc.

15   Device alarms can be generated in any desired manner, including with the use of proprietary or non-proprietary software located on a device itself or in other devices connected to the device for which the alarm is being generated to recognize and detect specific problems with the device and to generate an alarm with respect thereto. As indicated above, many smart devices are now being produced to create and

20   communicate device alarms.

Of course, there can be many different types of device alarms including, for example, failure alarms indicating that a failed or failing condition exists within a device, maintenance alarms indicating that maintenance of some type should take place, communication alarms indicating that a device is not communicating properly

25   or at all, advisory alarms, etc. A failure (e.g., a "failed") alarm indicates that a device has detected one or more conditions indicating that it cannot perform a critical function and, thus, requires maintenance immediately. Whenever the failed alarm condition is true, the integrity of the device is considered bad, which rolls up to the controller and causes the integrity of the controller node to which the device is

30   connected to be bad. On the other hand, a maintenance alarm indicates that a device

is able to perform critical functions but has one or more detected conditions that may lead to a failure if left unaddressed and, thus, the device should receive maintenance attention soon. A communication (e.g., a "not communicating") alarm becomes active when a device stops communicating. Whenever the not communicating alarm condition is true, the integrity of the device is considered bad, which causes the integrity of the controller node to which the device is connected to be bad. An advisory alarm indicates that a device has detected conditions that do not fall into the other alarm categories. Usually, an advisory alarm is an alarm provided by individual devices particular to the type of device, such as a flow meter tracking the variability of the flow signal. In this case, the device may recognize that a variability in some signal associated with the device is too high or too low, which means that something unusual has happened and requires investigation. Depending on the device, advisory alarms may require more or less urgent attention than maintenance alarms and, thus, users may set the priority of the advisory alarm lower than that of the maintenance alarm. Of course, failed, maintenance and advisory alarms may not be supported by every device and a single, catch all alarm, such as an "abnormal" alarm for generic devices may be used instead of the failed, maintenance, and advisory alarms resulting in two total alarms, i.e., not communicating and abnormal. Of course, other types of device alarms could be created or used instead of or in addition to the ones discussed above.

As described above, the alarm messages generated by Fieldbus devices do not include the types of information fields that are typically contained within process messages. In fact, the block identification, relative identification, and subcode fields are relatively cryptic and, thus, the numeric values in these fields do not provide useful information to a system user, operator, etc. Nor can these cryptic Fieldbus device generated alarm message fields be displayed on the display 69 by the alarm processing software 50 in a manner that is consistent with the display of process alarm messages.

To display device alarms in a format that is consistent with the display of process alarms, the alarm display and interface system described herein may enhance or extend device alarm messages prior to displaying these messages via the alarm

processing and display software 50. By way of example, a block within one of the Fieldbus devices 32-35 may generate a device alarm having the well known Fieldbus device alarm message format described above and may send this conventional alarm message over the bus 42 to the controller 12 via the I/O device 22B. The controller 12

5   may then append more information such as a manufacturer identification field, a device type field and a revision level field to the device alarm message based on the particular device from which the controller 12 received the message. As a result of system configuration activities, the controller 12 includes a device table that contains manufacturer identification information, device type information and revision level

10   information for each of the Fieldbus devices 32-35. By appending the manufacturer, device type and revision level information to each device generated alarm message, the controller 12 provides additional contextual information which, as described in greater detail below, may be used by the alarm processing software 50 to translate the conventional device alarm message fields into a displayed alarm message that is

15   consistent with the display of traditional process alarms.

The controller 12 sends the extended or enhanced device alarm messages via the bus 40 to the workstation 14 for processing by the alarm processing software 50. The alarm processing software 50 uses the appended manufacturer identification, device type and revision level fields of the extended device alarm message to select

20   one from a plurality of device alarm tables. The device alarm tables enable the alarm processing software 50 to map, translate or match each possible combination of enhanced alarm message field information (i.e., manufacturer identification, device type identification, revision level, block identification, etc.) to a set of displayable textual fields that contain the same types of information displayed in connection with

25   process alarms. For example, each of these tables may include, a textual message, an alarm type (e.g., HI, FAILED, MAINT, etc.), a description of the device sending the alarm, an alarm parameter, and any other alarm information that is typically provided by the alarm processing software 50 when displaying process alarms via the display 69. Additionally, the alarm device tables may include corrective action information

30   appropriate to the type of device and the nature of the alarm being generated by the

device. By way of example only, the corrective action information may indicate that recalibration of the device is needed, that some type of maintenance or repair should be performed on the device, that the device should be removed from service, etc.

The alarm processing software 50 and the device alarm tables may also enable the display of messages using one of a plurality of spoken/written languages. For example, a Japanese user may indicate to the alarm processing software 50 (via, for example, the computer 14) that alarm messages are to be displayed in Japanese. In response, the alarm processing software 50 may use an appropriate set of device alarm tables that enable the alarm processing software 50 to display Japanese language messages to the Japanese user. Of course, device alarm tables including messages in any number of different languages may be provided and each user may be given the option to select a preferred message display language.

Additionally, the alarm processing software 50 may insert the numeric information contained within the floating point field of the enhanced alarm message into the displayed device alarm information. By way of example, the alarm processing software 50 may recognize a special character within the message field of the device alarm tables that indicates whether or not and where a numeric value should be inserted when the message field is displayed in connection with a device alarm. Specifically, one or more of the message fields may include the characters "%V" to indicate that the numeric value within the floating point field of the device alarm message is displayed along with textual information contained in that message field. Furthermore, the relative position of the "%V" characters indicates precisely where within textual information or string the numeric value is to be inserted so that when viewed by a user, operator, etc. on the display 69, the numeric value can be read to have the proper meaning. Several alarms having message fields containing numeric values inserted therein are shown in Figs. 12-14, which are discussed in greater detail below. Of course, in some cases, message fields within some of the device alarm tables may not contain the characters "%V," in which case the alarm processing software 50 ignores any floating point field that may have been sent with the device alarm message.

Thus, the alarm processing software 50 completes the translation of enhanced device alarm messages into textual information that can be displayed using the user interface display 70 and the display 69 in a consistent and integrated manner along with process alarms. In other words, by extending or enhancing the device alarm messages within the controller 12 and then mapping these enhanced device alarm messages to textual fields containing alarm information that is preferably, but not necessarily, similar or identical in format (i.e., includes the same information fields) as displayed process alarms, device and process alarms may be displayed on the display 69 in a consistent manner. That is, from the perspective of a user, operator, etc. who is viewing process and device alarms on the display 69, all the alarms will include the same type of detailed information, thereby facilitating the integration of process and device alarms within a single display and enabling a user, operator, etc. to more easily manage multiple types of alarms.

Of course, the problem which causes some of the device alarms, such as communication alarms and failure alarms, may also cause one or more process alarms to be generated by the process control software. In particular, a communication alarm indicates that the communication connection with the device is broken and, thus, the process controller connected to the device does not know what is happening within the device. As a result, process control software, such as function blocks, expecting information from the device may generate process alarms indicating that the process loop is faulty.

Generally speaking, hardware alarms are alarms that indicate problems with a workstation, database, controller, I/O device or other hardware device, besides a field device, used within a process control system. Hardware alarms may indicate, for example, I/O outages or devices reporting problems with communication integrity. In particular, hardware alarms are used to alert users to failures of hardware components of the system and may appear in the operator interface as being associated with a node in which the failure was detected. Because hardware alarms may be used to notify operators and maintenance personnel of hardware failures within a node, hardware alarms augment known node integrity reporting systems. For example, if an eight

- 21 -

channel I/O card fails, software (e.g., the software 51 or 52 of Fig. 1) within the controller attached to the I/O device may report the I/O card failure as a hardware alarm. This is different from the past in which the operator may have had to deduce a failure of an I/O card based solely on the generation of a number of process alarms

5   caused by the disruption in the process control routines being performed using the faulty I/O card.

In one embodiment, each hardware component may generate two types of hardware alarms, including a "not communicating" alarm and a "failed" alarm. The not communicating alarm is generated when a device fails to communicate. The not

10   communicating alarm may be generated by the device having communication problems or may be generated by another device that expects communication from the device for which the alarm is generated and does not receive the communication after a time-out period, or may be generated by a device that polls the failed device for communication but does not receive an answer. On the other hand, the failed alarm

15   may become active when the component is communicating, but has detected one or more failures within the component (e.g., a failed channel in an I/O card). Failed alarms may be different based on the type of hardware device to which the alarm pertains. For example, for controllers, a failed alarm condition may result from a standby controller not being able to accept a switchover for a longer period than the

20   normal synchronization time, or may result when the controller has no configuration. For local application stations, a failed alarm condition may result from the application station having no configuration. For conventional I/O cards, a failed alarm condition may result from a failed channel, a standby not accepting a switchover for a longer period than the normal synchronization time, etc. For AS Interface I/O cards, a failed

25   alarm may result from a failed port, a port-device communication failure, a port-device abnormal status, a standby not accepting a switchover for a longer period than the normal synchronization time etc. For Fieldbus H1, Profibus I/O, serial I/O cards and the like, a failed alarm may indicate that a specific port has failed, that there is a port-device communication failure, that there is a port-device abnormal status, that a

30   standby card is not operating properly, etc.

Because hardware failures can also result in device alarms and process alarms, the effective priority of all device alarms for all devices communicating through a failed hardware component may be forced to a low priority level, such as priority level three, by the alarm processing unit 64. This will effectively remove all device and

5 process alarms from the user display for devices that are affected by this hardware failure (such as a port failure) until the hardware unit returns to a communicating state.

As will be understood, device and hardware alarms are used to alert users to failures and other conditions detected in devices connected to the system, such as

10 smart devices connected to controllers via various technologies such as Fieldbus, HART, Profibus, etc. and other hardware, such as I/O devices, controllers, application workstations, etc. While the foregoing discussion describes the preferred user interface elements relating to device alarms and hardware alarms using Fieldbus device alerts/alarms, the expectation is that device and hardware alarms from other

15 smart I/O and control systems could be used as well.

As noted above, the different categories of alarms, including process alarms, device alarms and hardware alarms, are sent to and received by the alarm processing software 50 for potential display on the display device 69 in some convenient message format, such as the format discussed above. The actual alarms that are presented on

20 the display device 69 are determined by the scope of access of the workstation and operator and by the filter settings of the filter 68, which can be configured to display alarms using any desired criteria, such as the alarm category (e.g., process, device or hardware), the alarm priority, etc. Using this software, different categories of alarms are integrated or displayed within the same interface to provide an operator more

25 information pertaining to the faulty operation of the process control system 10 in contrast to the past, where an operator could only view process alarms and had to use these process alarms to determine if device or hardware failures were underlying (i.e., caused) the process alarms. With the integrated alarm displays described herein, an operator can view actual device and hardware alarms on the same screen or display

30 device as the process alarms and can treat each of the alarms in the same manner,

which enables the operator to determine more quickly and easily if one or more process alarms are a result of a faulty device or hardware, etc. Likewise, the user can interact with and manage the device and hardware alarms in the same manner that the user manages process alarms so that reacting to each of these categories of alarms requires the same types of user functions.

Of course, there are many ways in which the different categories of alarms may be displayed in an integrated manner on a user interface. In one embodiment, the process, device and hardware alarms are treated similar to the way in which process alarms have traditionally been treated on a display. As a result, an operator can acknowledge or suppress device and hardware alarms in the same manner in which process alarms are acknowledged or suppressed. Likewise, device and hardware alarms may be displayed in a manner that indicates the type, priority, name, section of the process, state, etc. of the alarm. Also, a primary control display associated with an alarm may be presented to the user, with the primary control display being a display that helps the user understand or recognize the source of the alarm or functionality of the hardware or software element associated with the alarm, such as the module, process loop, device, node, area, etc. for which the alarm was generated or with which the alarm is associated. A primary control display may be, for example, a physical picture of a device, a digital picture or drawing of the room or the area in which a device is located, other information associated with the device such as part of a plant drawing, schematic or conception drawing illustrating the connections between the device in the plant during implementation, etc. Primary control displays for alarms can be created by users and may, for example, be oriented to modules (for process alarms), to devices (for device alarms) and to nodes (for hardware alarms) or to areas or sections of the plant associated with the alarm. The primary control displays may be geared to different functions also. For example, process alarm primary control displays may be oriented to process operation functions, device alarm primary control displays may be oriented to field device maintenance functions and hardware primary control displays may be oriented to node maintenance functions. Primary control displays for hardware alarms may be, for example, pictures of where the controller is

- 24 -

located, schematics of the controller's I/O hardware with all hardware alarm statuses indicated, buttons to navigate to the unit overview or primary control displays that a controller is supporting, maintenance procedure checklists, etc. Likewise, primary control displays for device alarms can be created by users and may, for example, be

5   oriented to device maintenance functions. The primary control displays may be stored in the database 66 (Fig. 2) and may be accessed and presented on the display 69 when an alarm using that primary control display is selected. Of course, the same or different primary control displays may be used for different alarms and primary control displays.

10   In one embodiment, the integrated alarm information may be provided to a user on a display in the form of an alarm banner at, for example, an edge of a display screen. Referring now to Fig. 3, an alarm banner 73 is located on the bottom of a screen 71. The alarm banner 73 includes a first line that displays indications of various alarms that have been generated by the process control system 10 and that

15   have passed through the filter 68 to the display. At least one of the alarms indicated in the alarm banner 73 may be associated with the portion of the process control system 10 depicted in the main part of the screen 71. The specific alarms displayed in the alarm banner 73 and the order of these alarms are determined according to the filter settings of the filter 68. Generally speaking, the highest priority alarms that have not

20   been acknowledged or suppressed will be displayed first, with the next highest priority arms being displayed next, and so on. In the exemplary screen of Fig. 3, the highest priority alarm 74 is a process alarm illustrated as being associated with a PID101 control routine. The alarm 74 is displayed in red to illustrate that its priority is critical. On the second line of the alarm banner 73, an alarm information field 76

25   displays alarm information associated with the alarm in the alarm banner 73 that is currently selected. In the example of Fig. 3, wherein the alarm 74 is selected, the alarm information field 76 illustrates that the alarm 74 was generated on Friday at 12:52:19, is associated with the "tank 16 level control," has a designation or name of PID101/HI_HI_ALM, has a high, high priority and is a critical alarm. If the alarm 74

30   is flashing, this means that the alarm is not acknowledged, while a constant (non-

flashing) alarm indication in the alarm banner 73 means that the alarm has been acknowledged by some operator or user. Of course, other types of alarm information could be displayed within the alarm information field 76.

Also, the other alarm indications in the alarm banner 73, such as the alarm indication 78, may be yellow, purple, or any other color to indicate other levels of seriousness or priority associated with the alarm. When another alarm is selected, such as the alarm 78, 80, 81 or 82, alarm information pertaining to that alarm may be displayed in the alarm information field 76. When viewing an alarm in the alarm banner 73, the user can acknowledge the alarms and alert maintenance or engineer personnel to take the appropriate actions to correct the condition led to the alarm or, alternatively, could take other steps such as resetting certain set points to alleviate the alarm condition. Of course, when used to display only process alarms, the display of Fig. 3 is similar to a well known operator display provided in the Delta-V control system.

As indicated above, by selecting one of the alarms in the alarm banner 73 such as the alarm 74, a primary control display for that alarm is presented in the screen 71. In particular, as shown in Fig. 3, the main body of the screen 71 includes a primary control display or depiction of pertinent hardware associated with a particular alarm (a selected alarm) within the process control system 10. In the example of Fig. 3, the hardware includes three tanks with various sensors attached thereto, all of which are interconnected by various valves and fluid flow lines. This hardware depiction is a representation of the equipment within a portion of the process control system 10 and provides certain information about the operation of some of the equipment, such as certain values or parameters associated with the tanks, sensors etc. Of course, some of this information may be provided by configuration information in the database 66 and signals from the sensors in the process control system via the controllers 12 and Ethernet connection 40. In this case, such information is sent up through the communication layer 62 and is provided to the user display interface 70 via any known or desired software.

Also, as illustrated in Fig. 3, a face plate 72 depicting a "virtual instrument" for a PID control unit (module) is illustrated as additional information for one of the alarms (in this case, the process alarm 74) within the alarm banner. The face plate 72 provides further information pertinent to the selected process alarm and identifies the

5 name of the control unit (the module PID101) and certain settings or parameters associated with that module. The generation of such a pictorial description of the process is now used for process control alarms and is known in the art and, thus, will not be described in detail. In fact, any other desired pictorial or non-pictorial description of part of or the entirety of the process control system 10 may be displayed

10 on the screen 71 to enable a user, such as an operator, to view the operational functions or hardware functions of any part of the process control system 10. The displays, of course, may depict or otherwise represent individual hardware units, related groups of hardware, block diagrams or other diagrams of portions or areas of plants, etc.

15 In the past, displays such as the display of Fig. 3 have provided the capability of displaying process alarms and were used by operators to view process alarms, i.e., alarms associated with the software (such as function blocks) controlling the process operation. In contrast, the alarm display and interface software 50 described herein provides information about other categories of alarms including device and hardware

20 alarms in conjunction with process alarms, if the operator has set the filter 68 to view such alarms.

Referring now to Fig. 4, a further integrated alarm display having various categories of alarms presented on the alarm banner 73 is illustrated. In particular, in Fig. 4, the first and highest priority alarm 90 is a device alarm associated with a valve

25 101, the second highest priority alarm 92 is a hardware alarm associated with a controller 101 and the third alarm 94 is a process alarm associated with the process control module PID 101. The alarm information for the device alarm 90, which is selected in Fig. 4, is illustrated in the alarm information field 76. The information for the device alarm 92 includes event information pertaining to the time at which the

30 alarm was generated, the priority and name of the device along with information about

the criticality and priority of the alarm. Likewise, the primary control display for the valve 101 is illustrated on the screen 71 (which, incidentally is the same primary control display as the process alarm 74 of Fig. 3). Also, a face plate 98 for the valve 101 is illustrated. The face plate 98 is one designed for a device alarm and will be

5   discussed in more detail below in connection with Figs. 8 and 9.

An exemplary alarm banner information display for a process alarm is illustrated in Fig. 5. The alarm banner information shown in Fig. 5 is similar to that which has been provided for process alarms in the past. An exemplary alarm banner information display for a device alarm (FV-101) including the alarm information field

10   therefor is illustrated in Fig. 6. In this display, "FV-101" is a user configured name of the device; "Reactor 1 inlet valve" is a user configured description associated with the device; and "FAIL_ALM" is a non-configurable name of the device alarm parameter. The alarm parameter names may be, for example, COMM_ALM for "not communicating" device alarms, FAIL_ALM for "failed" device alarms,

15   MAINT_ALM for "maintenance" device alarms and ADVISE_ALM for "advisory" device alarms. Also, in Fig. 6, "FAILED" is the a-configurable alarm word, which has been selected from the set of potential alarm words "COMM, FAILED," "MAINT," and "ADVISE." "CRITICAL" is the alarm priority word and "I/P Feedback limit: 103.47" is a non-configurable string that is determined based on the

20   device, revision, and the information available from the device as described in greater detail above. For alarms detected from Foundation Fieldbus alert notification, the string depends on the data from the alert message and may optionally include one numeric value inserted in the string as described above.

Because Foundation Fieldbus devices are not expected to report any other

25   conditions contributing to an alarm while the first condition in the alarm is active, the message that appears in the alarm banner describes the first condition detected. The timestamp shown for the alarm (and that is used in ordering of alarms in alarm banner and alarm summary displays) is the time that the alarm first went to an active state.

Examples of typical types of device alarm messages are NVM write count

30   limit, 100001, Output block time-out, Pressure derivative limit, pressure high limit,

- 28 -

Pressure low limit, I/P Feedback limit, Temperature high limit, Temperature low limit, Travel deviation, 25.67 servo units, Travel high limit, Travel high-high limit, Travel low limit, Travel low-low limit, Travel accumulation limit, cycle count limit, Drive failure, etc.

5   Likewise, an exemplary alarm banner for a hardware alarm is illustrated in Fig. 7. In this display, "CTLR1" is a user configured name of the node; "Room 4, cab, 3, pos 2" is a user configured description associated with the node; and "CARD04_FAIL" is a non-configurable name of an alarm parameter. The alarm parameter names may be, for example, NODE_COMM for "not communicating"

10   (standby) node hardware alarm, NODE_FAIL for "failed" (standby) node hardware alarm, CARDxx_COMM for I/O card "not communicating" hardware alarms, CARDxx_FAIL for I/O card "failed" hardware alarms, etc. Further, in Fig. 7, "FAILED" is the non-configurable alarm word selected, for example, from the potential alarm words, COMM for "not communicating" node or I/O card hardware

15   alarms and FAILED for "failed" node or I/O card hardware alarms. "CRITICAL" is the alarm priority word and "Channel 7 failed" is a description determined based on the condition in the node or I/O card that is contributing to the alarm. The message that appears in the banner describes the last condition change (i.e., the condition either going active or inactive) detected while the alarm is in an active state. The timestamp

20   shown for the alarm, which may be used in ordering of alarms in alarm banner and other alarm displays, is the time that the alarm first went to an active state.

From Figs. 5-7 it can be seen that the information presented in the alarm banner information portion is similar for each of the different categories of alarms that are being displayed or indicated on the user interface. Likewise, the alarm indications

25   share the same display features such as blinking, color, etc. based on the priority, active, acknowledged status of the alarm, despite the category of the alarm. In this manner, alarms of different categories are integrated within the display screen, e.g., are shown together and are treated similarly based on the relevant parameters of the alarms.

Referring now to Figs. 8-10, exemplary face plate displays for device and hardware alarms are illustrated. In particular, Fig. 8 depicts a face plate display for a device FV-101, illustrating the different alarm categories available for this device, which in this case are NO COMM, FAILED, MAINT and ADVISE alarms. For each

5   of these categories of alarms, an ack button (acknowledgment button), an enabled button (EN) and a suppressed button (SUP) are provided. Using these controls, the operator can acknowledge, enable, disable and suppress the different kinds of alarms for this device. Likewise, priority adjustment controls enable a user to adjust the priority of the alarms for this device. The Details button may be used to call up other

10   applications can be used to access this device or to find out other information about the device. Any known prior art applications can be used for this purpose and these applications will not be described herein. Still further, active alarms may be indicated using color and blinking features, similar to the alarm indications in the alarm banner. Fig. 9 illustrates a primary control display for a generic device which has two

15   categories of alarms, namely, a NO COMM and an ABNORMAL alarm.

If desired, using the alarm banner buttons to call up a primary control display for a device alarm will also bring up a face plate display, such as shown in Fig. 8 or 9, indicating the most important active device alarms for that device. For example, up to a certain number (e.g., five) of the most important device alarms in the device may be

20   indicated using consistent color and blinking as the representation in the alarm banner. It is also possible, using the displays of Figs. 8 and 9, to acknowledge, disable or adjust the effective priority of all the device alarms in the device. As discussed in more detail below, it is also possible to open a display showing a summary of all active device alarms in the device.

25   Similarly, Fig. 10 illustrates an exemplary face plate for a hardware device, which in this case, is a controller named CTLR1. The face plate shown Fig. 10 illustrates the different alarms available, such as CTRL_FAIL, CARD1_FAIL, etc. that indicate a portion of the node and the alarm category (failure or communication alarm). Again, if desired, the hardware face plate may show the most important active

30   hardware alarms for that node. For example, up to a certain number (e.g., five) of the

most important hardware alarms in the node may be indicated using consistent color and blinking as the representation in the alarm banner. It is possible, using the face plate display, to acknowledge, disable or adjust the effective priority of all the hardware alarms in the node. It is also possible to open a display showing a summary

5   of all active hardware alarms in the node by pressing the summary button on the face plate display of Fig. 10.

Generally speaking, in one embodiment such as that illustrated in Figs. 3 and 4, the order that the alarm indications appear in the alarm banner 73 follows a set of ordering rules which may or may not be reconfigured or changed dynamically by a

10   user. One exemplary set of such rules is as follows: (1) unacknowledged alarms appear before acknowledged alarms; (2) for alarms with the same acknowledgment status, alarms which are currently active appear before those which have cleared (i.e., before being acknowledged); (3) for alarms with the same acknowledgment and active status, alarms with higher priorities appear before alarms with lower priorities.;

15   and (4) for alarms with the same acknowledgment, active status and priority, alarms which occurred more recently appear before those which occurred longer ago.

In one embodiment, different kinds of information may be available for display about each active alarm in the alarm information banner 73 including, an alarm identifier ("<container name>/alarm parameter name"), a unit name, a time of

20   occurrence, a description associated with the <container name>, an alarm word and a priority word. The container name could be, for example, a module name for process alarms, a device name for device alarms and a node name for hardware alarms.

Preferably, users can configure alarm priorities so that a <container name> appears at most once in the alarm banner, so that a unit name (if any) appears in the

25   alarm banner buttons instead of the <container name>, and so that the unit name appears at most once in the alarm banner. It may also be possible to configure the alarm banner 73 to filter out alarms by category, by type or priority threshold (i.e., alarms below the threshold are not shown). Conceptually, the configuration could be accomplished using a window such as, for example, the window illustrated in Fig. 11,

30   wherein a user can select the alarm category to be displayed (e.g., process, hardware

- 31 -

or device alarms) and the priority level for the alarms within each category. These settings are stored in the filter settings and used by the filter 68 to control which alarm indications are displayed to the user via the alarm banner 73. Of course, the operator or user may be able to set the filter settings to filter out any of the types, categories,

5   priorities, etc. of alarms at different times to customize the alarm banner as desired.

With the integrated alarm display banner 73, such as that illustrated in Figs. 3 and 4, the screen 71 provides a single user with the information pertaining to different categories of alarms in essentially a uniform format which, in turn, enables a user to more readily understand the problems within the process control system 10 which may

10   be causing process control errors or alarms. This, in turn, may help the user determine what devices or settings need to be altered, fixed, replaced or maintained to restore the process 10 to desired working conditions. The integrated display 71 is also very useful in the case in which a single person, such as an operator, also performs the functions of maintenance or engineering personnel in a process control system, which

15   is generally the case in many small process control systems. In this way, the user does not have to view different displays or look at different databases to determine different types of errors within the system to diagnose a problem.

When the user chooses to view only process alarms, the user, in effect, converts the interface into a system in which only process control alarms are available

20   to the user and can perform the typical functions associated with operator personnel. Nevertheless, alarms are still being received and stored in the database 66 in case the user decides to switch back, at some point, to view more categories of alarms. Still further, the filter 68 allows a user to view different categories of alarms and, thus, enables the user to view all of the alarms when the number of outstanding alarms is

25   not too great. At this time, the user may try to determine if the cause of some process alarms is a result of one or more device or hardware failures, as evidenced by received hardware and device alarms. However, when a major problem occurs that causes many alarms to appear, the user can choose not to view certain categories of alarms, such as device or hardware alarms, or priorities of alarms and, thus, the user can focus

30   solely on the process alarms for which the user has primary responsibility. In this

- 32 -

manner, the user can hand off process, device or hardware alarms to other users who may use a version or copy of the software interface 50 at a different workstation to display only those alarms. As a result, a user can view the alarms that may be helpful to their main function but, when too many alarms are being displayed, the user can cut down the number of alarms by changing the filter settings thereby preventing the user from becoming overwhelmed by too many alarms.

Besides or in addition to an implementing an alarm banner, the alarm display and interface software 50 may include software that generates alarm summary objects (also referred to herein as windows, displays or controls) for the user, including, for example, active alarm summaries and suppressed alarm summaries. Active alarm summary displays may be designed using an alarm summary control, which provides additional filtering, sorting, scrolling and context actions for summarizing the alarms present in the system. Such a summary control software module 110 is illustrated in Fig. 2. The display generated by the alarm summary control 110 may be configured to occupy different positions and portions of the display area. For example, configurable widths may range from the full width of the display area (in the supported display resolutions) to about half of the full width. Configurable heights may, for example, range from the full height of the display area (excluding the framework for navigation buttons and alarm banner) to about half of the full height. Thus a full screen alarm summary, two half screen alarm summaries (both side by side and over under) and four quarter size alarm summaries may be supported in one display. Of course, other sizes and geometries could be used as well.

The alarm summary control 110 may be configured to create a display that shows any of the properties of an active alarm, including, for example, a container name (module/node/device), an alarm identifier (path) (<container name>/<alarm parameter name>), a description for <container name>, a unit name (if any), a priority word, a latched alarm state word (alarm word or "OK"), a current alarm state word (alarm word or "OK"), a category name, a time of occurrence, a time of last state/priority change, an alarm message string, a source node name, etc.

Other properties available for active alarms used to control the presentation or for display navigation include an alarm category value (1 = process, 2 = hardware, 3 = device), an area number (0...99), a priority numeric value (4...15), an acknowledgment state value (1 = unacked, 0 = acked), a latched alarm state value (>0 if active), a

5   current alarm state value (>0 if active), a faceplate display name for <container name>, a primary control display name for <container name>, a detail display name for <container name>, etc.

An example of an active alarm summary display is illustrated in Fig. 12. As shown in Fig. 12, the alarm summary display includes event information about the

10   time of the alarm, the unit in which the alarm occurred (if applicable), the alarm parameter, a description of the alarm, the type of alarm, a message associated with the alarm and the priority of the alarm. Of course, this and/or other summary information may be provided in any other format, if desired. Fig. 13 illustrates the same alarm summary information having colors used to highlight the alarms based on priority.

15   Thus, high priority (critical) alarms are highlighted in, for example, red, medium alarms, such as advisory alarms, are highlighted in yellow and other alarms are not highlighted. Fig. 14 illustrates the use of different colored highlighting to highlight alarms according to alarm category. In Fig. 14, device alarms are highlighted. Of course, the use of highlighting or other indicia may be used to mark alarms within the

20   alarm summary display according to any desired criteria, and marking or highlighting of different sections of the alarm summary display may be done according to different criteria. Thus, the priority field may be highlighted according to the priority, the alarm description and alarm parameter may be highlighted according to alarm category, etc.

25   In any event, each field to be displayed in an alarm summary display produced by the alarm summary control 110 may have any combination of the following presentation attributes configured. A background blinking based on acknowledgment state, a background color based on either priority value (using same color rules as alarm banner) or alarm category, and a hash overlay based on current alarm state. The

30   alarm summary control may be configured to show alarms in any desired order, such

as ack state / current alarm state / priority / time of occurrence (alarm banner order), time of occurrence (newest to oldest), time of last state/priority change (newest to oldest), alarm identifier (<container name>/<alarm parameter name>) (alphabetical), etc.

5 The alarm summary control 110 may be configured to show alarms with zero or more of the following filter criteria, (1) current alarm banner filter settings, (2) alarm category and priority thresholds (any combination of process, hardware, and device alarms and priority thresholds for each), (3) area name, (4) unit name, (5) <container name>, (6) acknowledgment state (0 or 1), (7) current alarm state > 0, (8) 10 time of occurrence < x days, y hours, z minutes relative to now, or (9) time of occurrence > x days, y hours, z minutes relative to now.

When an alarm summary display is presented on the display 69, a user holding the proper security keys may change the filter criteria such as alarm category and priority thresholds (e.g., any combination of process, hardware, and device alarms and 15 priority thresholds for each), acknowledgment state (0 or 1), current alarm state > 0, time of occurrence < x days, y hours, z minutes relative to now or time of occurrence > x days, y hours, z minutes relative to now, etc.

Alarm summary displays may remain open or created for any length of time and, preferably, when the presentation attributes, alarm order, or filter criteria for an 20 open alarm summary display are changed dynamically by a user, the configuration of the control 110 is not changed and the changes are discarded when the alarm summary display is turned off. Of course, there are no limits (other than CPU resources) on the number of alarms that may appear in an alarm summary control.

Also, the alarm summary control 110 may be used to perform operations with 25 respect to an alarm such as acknowledging an alarm (normal parameter/field security rules apply), suppressing an alarm (normal parameter/field security rules apply), calling up the display for the <container name> holding an alarm, calling up the primary control display for the <container name> holding an alarm, calling up the detail display for the <container name> holding an alarm, launching or bringing up a 30 diagnostic application with the appropriate module, device or node selected, i.e., a

module for process alarms, a device for device alarms or a node (or a card) for hardware alarms.

If desired, the alarm summary display created by the control 110 may have an "Ack all" button which, when enabled for an open alarm summary, may be used to
5   acknowledge all unacknowledged alarms that are included in this alarm summary (i.e., those which passed the workstation-wide and operator-wide alarm scope controls and which passed the current filter criteria for this alarm summary), including unacknowledged alarms that are in this summary but are not currently displayed. The effect is the same as individually selecting each unacknowledged alarm in the alarm
10   summary and acknowledging that alarm.

Still further, the alarm display and interface software 50 may include a suppressed alarm summary control 112 which provides a suppressed alarm summary display. This display can be designed around a summary control which provides additional filtering, sorting, scrolling, and context actions for suppressed alarms. The
15   characteristics of suppressed alarm summary controls 112 are similar to that for active alarm summary controls 110 described above, except that the suppressed alarm summary control 110 may also be configured to show any of the following properties of an active alarm including, <container name> (module/node/device), the alarm identifier (path) (<container name>/<alarm parameter name>), a description for
20   <container name>, a unit name (if any) and a time of suppression.

Additionally, unit or plant area alarm displays can be generated to show all or a group of alarms associated with a unit or an area or other logical grouping of hardware within the process control system 10. A plant area display can provide a list of all plant area names and, for each plant area, may provide a control that shows if
25   the area is within the scope of the workstation and user (i.e., if alarms can be turned on or off), the current state of alarms being turned on or off. These controls can also be used to change the state, i.e., to turn areas or units on or off. The plant area display can also show alarm counts for each area (the counts are 0 if the area is turned off) such as counts for active/unacknowledged process, device and hardware alarms,
30   active process, device and hardware alarms, and suppressed process, device and

- 36 -

hardware alarms. The display may also provide a means to open an alarm summary control for any plant area. The default alarm summary control that appears may show all alarms in the area (process, hardware, and device) in, for example, ack state / current alarm state / priority / time of occurrence (alarm banner) order. Also, to allow

5   users to configure different alarm banner displays or alarm summary displays for different workstations, it is possible to configure the name of the alarm banner display or alarm summary display on a per workstation basis. Of course, all of these displays may have different categories of alarms displayed therein in similar manners.

If desired, the alarm processing unit 64 keeps a set of alarm counts for the

10   alarms that pass the workstation-wide alarm scope controls and makes these counts available for display. Alarm counts may be kept, for example, for the active/unacknowledged process, device and hardware alarms, for the active/acknowledged process, device and hardware alarms, and for the suppressed process, device and hardware alarms. Preferably, these counts are not affected by the

15   alarm banner alarm category and priority threshold filter settings. For example, device alarms can be turned off in the alarm banner, but the counts for device alarms may still be accurate, given the workstation-wide alarm scope controls.

Still further, the software 50 may display an alarms parameter that is used to show the five most important alarms in a device (or a node) to be supported for each

20   device or node which has device or hardware alarms enabled. This functionality makes it possible to acknowledge, disable or affect the priority of all device alarms for a device or hardware alarms for a node via a single parameter/field. A browser for the system may also browse device alarms and hardware alarms. The alarm interface software 50 may support the ability to read/write alarm parameter fields of individual

25   device alarms from control modules that run in the same node.

As will be understood, the alarm display and interface software 50 combines process, device and hardware alarms in a manner so that these different categories of alarms share similar behavior and may share, for example, alarm priority words, color coding, acknowledgment behavior, etc. If desired, individual alarm acknowledgment

30   and suppression can function identically for all of these alarms. Also, an alarms

parameter (like the one currently provided in each module for process alarms) may be used to show the five most important alarms in a device or node. This parameter can also be used to acknowledge, disable and adjust the effective priority of all hardware alarms for a node or device alarms for a device via a single parameter/field written.

5   Because device alarms and hardware alarms belong to or are associated with plant areas, turning on/off alarms for an area can affect the device and hardware alarms for that area.

As currently provided with process alarms, users may place links to fields of device and hardware alarm parameters in displays, such as the display 71 and a

10   browser may be used in building displays that support browsing to device and hardware alarms. Also, if desired, a user may read/write alarm parameter fields of individual device and hardware alarms from control modules. The parameter browser used in creating module configurations supports browsing to device and hardware alarms.

15   Of course, process, device and hardware alarms have a category parameter that distinguishes them from each other, and this parameter is readable by the alarm processing unit 64. As a result, the alarm banner and alarm summary displays may be configured to include/exclude any of these process, device, hardware alarms, and may be configured to give the process, device and hardware alarms distinct appearances to

20   enable a viewer to easily distinguish between these different categories of alarms.

If desired, an event chronicle database may be stored in a workstation or other device at the node which runs the software 50 or any other node to capture the state changes of process, device and hardware alarms and to store these changes in a manner that allows the history of the changes to be recovered. The event chronicle

25   may, for example, store the current alarm state as one of SUPPRESSED, DISABLED, INACT/ACK, INACT/UNACK, ACT/UNACK or ACT/ACK. A default state for a device and hardware alarm is preferably INACT/ACK. When a download occurs that creates a device or hardware alarm, no initial state change record is generated to record the alarms "initial" INACT/ACK state. However, subsequent state changes

30   detected would generate the appropriate alarm state change event record.

A process history view may use standard filter/sorting features to quickly find all process, device or hardware alarm state changes, specific categories of alarms from a single node, from a specific plant area, etc. Exemplary event chronicle state change record sequences for process and device alarms be as depicted in Fig. 15. Here, the category word that appears for all alarm state change records is fixed according to the category of alarm. The message shown in the Desc2 field describes the condition within the alarm that changed. The word that appears in the Desc1 field indicates the state of the condition described in the Desc2 field: the normal alarm word if the condition is active; "OK" if the condition is inactive. Of course, similar types of entries for hardware and other categories of alarms may also be provided in the event chronicle and other types of information for recording manipulations of the alarms may also be stored in any desired manner in the event chronicle database.

The device or hardware alarms are part of devices and hardware nodes as these are created in the configuration database. Thus, the alarms are created as part of the configuration in a manner similar to the manner in which other aspects of devices and nodes are created. When the configuration is created in the system, all device and hardware alarm behaviors for the device and hardware nodes may be set to be "off" by default. While these alarms are off, alarm parameters are unavailable for the device or hardware units so that these alarms do not appear as part of the configuration for the device or hardware unit and these alarms are not accessible via configuration parameter browsers. That is, the devices and hardware alarms do not appear in the system browser as being associated with the field devices and nodes. Furthermore, there is no run-time support for device and hardware alarms and all device and hardware alert communications including any parameter polling used to maintain state information supporting device and hardware alarms is halted.

However, a user can enable device and hardware alarms using, for example, an on/off or checkbox style property of the device or hardware. When the alarms are enabled, the device or hardware name must conform to configuration system naming rules so that these names can be used to address alarm information or messages from the operator interface applications and other applications. Specifically, the name must

conform to name rules, which may include, for example, maximum character length and character restrictions and must share the name space with modules, nodes, areas, and DST names or other names. Generally, if the device or hardware name does not conform to name rules used in the system, device and hardware alarms cannot be

5   enabled and the user is notified. The description string entered for the device or hardware appears in the operator interface to help describe active alarms.

New configurable properties for configuring a device or a hardware unit may include the name of the faceplate display for this device or node and the name of the primary control display for this device or node. When device or hardware alarms are

10   enabled for a device or hardware entity, an alarm parameter appears in the configuration system for each possible alarm that the system supports for that device or hardware entity. Information provided via the configuration system may include the manufacturer, type, and revision of the device or hardware unit which determines the number of the alarms that are available, their names, as well as the default values

15   for the configurable properties. When alarms are enabled for a device or a hardware entity, the configuration system enables the user to see and change (if configurable) the name, alarm word, type and priority properties of each device alarm. Priority is generally configurable using the priorities that are currently defined using similar technique as is supported for alarms in modules. For example, default numerical

20   values for priority may be COMM_ALM - 15 ("CRITICAL"), ABNORM_ALM - 11 ("WARNING"), FAILED_ALM - 15 ("CRITICAL"), MAINT_ALM - 7 ("ADVISORY"), ADVISE_ALM - 7 ("ADVISORY").

To support device and hardware alarms, it will be possible to configure an association between a node and a plant area. This applies to all node types including

25   controllers and various workstations, and local and remote nodes. New nodes may be associated with plant area 0. Nodes imported with no node association are associated with plant area 0. Device alarms are implicitly associated with the plant area that their node is associated with.

While the alarm display and interface software 50 has been described as being

30   used in conjunction with Fieldbus and standard 4-20 mA devices, it can be

implemented using any other external process control communication protocol and may be used with any other types of controller software. Although the alarm display and interface software 50 described herein is preferably implemented as software, it may be implemented in hardware, firmware, etc., and may be implemented by any

5   other processor associated with the process control system 10. Thus, the routine 50 described herein may be implemented in a standard multi-purpose processor or using specifically designed hardware or firmware as desired. When implemented in software, the software routine may be stored in any computer readable memory such as on a magnetic disk, a laser disk, or other storage medium, in a RAM or ROM of a

10   computer or processor, etc. Likewise, this software may be delivered to a user or a process control system via any known or desired delivery method including, for example, on a computer readable disk or other transportable computer storage mechanism or over a communication channel such as a telephone line, the Internet, etc. (which are viewed as being the same as or interchangeable with providing such

15   software via a transportable storage medium).

       Thus, while the present invention has been described with reference to specific examples, which are intended to be illustrative only and not to be limiting of the invention, it will be apparent to those of ordinary skill in the art that changes, additions or deletions may be made to the disclosed embodiments without departing

20   from the spirit and scope of the invention.